

# Spectral Clustering via Graph Filtering: Consistency on the High-Dimensional Stochastic Block Model

Muni Sreenivas Pydi

MUNISREENIVAS@CSA.IISC.ERNET.IN

Ambedkar Dukkipati

AD@CSA.IISC.ERNET.IN

*Department of Computer Science and Automation*

*Indian Institute of Science*

*Bangalore 560012, India*

## Abstract

Spectral clustering is amongst the most popular methods for community detection in graphs. A key step in spectral clustering algorithms is the eigen decomposition of the  $n \times n$  graph Laplacian matrix to extract its  $k$  leading eigenvectors, where  $k$  is the desired number of clusters among  $n$  objects. This is prohibitively complex to implement for very large datasets. However, it has recently been shown that it is possible to bypass the eigen decomposition by computing an approximate spectral embedding through graph filtering of random signals. In this paper, we prove that spectral clustering performed via graph filtering can still recover the planted clusters consistently, under mild conditions. We analyse the effects of sparsity, dimensionality and filter approximation error on the consistency of the algorithm.

**Keywords:** Spectral Clustering, Stochastic Block Model, Graph Filtering

## 1. Introduction

Detecting communities, or clusters in networks is an important problem in many fields of science (Fortunato, 2010; Jain et al., 1999). Spectral clustering is a widely used algorithm for community detection in networks (Von Luxburg, 2007) because of its strong theoretical grounding (Ng et al., 2002; Shi and Malik, 2000) and recently established consistency results (Rohe et al., 2011; Lei et al., 2015). Spectral clustering works by relaxing the NP-hard discrete optimization problem of graph partitioning, into a continuous optimization problem. As a first step, one computes the  $k$  leading eigenvectors of the graph Laplacian matrix, that gives a  $k$  dimensional 'spectral' embedding for each vertex of the graph. In the second step, one performs  $k$ -means on the  $k$  dimensional dataset to retrieve the graph clusters.

However, the eigen decomposition that is required to extract the top  $k$  eigen vectors of the graph Laplacian is very hard to compute for large datasets. Several approximate algorithms have been proposed to overcome this problem via Nyström sampling (Fowlkes et al., 2004; Li et al., 2011; Choromanska et al., 2013). While these methods do not skip the eigen decomposition step, they reduce its complexity via column sampling of the Laplacian matrix. Another class of methods use random projections to reduce the dimensionality of the dataset while obtaining an approximate spectral embedding (Sakai and Imiya, 2009; Gittens et al., 2013). With the emergence of signal processing on graphs (Shuman et al., 2013), there has been the development of techniques based on graph filtering that can side-step the eigen decomposition altogether (Ramasamy and Madhow, 2015; Tremblay et al.,

2016b,a). While many of these approaches have been shown to work fairly well on real and synthetic datasets, a rigorous mathematical analysis is still lacking.

In this paper, we consider a variant of the compressive spectral clustering algorithm as proposed by Tremblay et al. (2016b) which uses graph filtering of random signals to compute an approximate spectral embedding of the graph vertices. For a graph with  $n$  vertices and  $k$  clusters, the algorithm proceeds by calculating a  $d$  dimensional embedding for the vertices of the graph, where  $d$  is of the order of  $\log(n)$ . This ‘compressed’ embedding acts as a substitute for the  $k$  dimensional spectral embedding of the spectral clustering algorithm as given in Algorithm 1. The embedding is obtained by filtering out the top  $k$  frequency modes for  $d$  number of random graph signals. Subsequently,  $k$ -means is performed on the new embedding, just as in case of spectral clustering. While the algorithm may speed up the  $k$ -means step in the case of high-dimensional models where the number of clusters  $k$  grows with  $n$ , the most significant gains in computation are obtained by avoiding the eigen decomposition of the  $n \times n$  graph Laplacian matrix. Although ideal filtering of the top  $k$  frequency modes of a graph signal requires computing the top  $k$  eigenvectors of the graph Laplacian, Tremblay et al. (2016b) show that a polynomial approximation of the ideal filter can be used to skip the eigen decomposition.

## Contributions

We analyse the spectral clustering algorithm performed via graph filtering (Algorithm 2) using the stochastic block model. We derive a bound on the number of vertices that would be incorrectly clustered with the algorithm, and prove that the algorithm can consistently recover planted clusters from a stochastic block model under mild assumptions on the sparsity of the graph and the polynomial filter approximation used to compute the spectral embedding. In the process, we also characterize the filter approximation error and the error in the approximate spectral embedding, that are inherent to the algorithm.

For our analysis, we specifically consider the high-dimensional stochastic block model that allows for the number of clusters  $k$  to grow faster than  $\log(n)$ . This is because the computational gains of Algorithm 2 are more apparent in the high-dimensional case. In proving the weak consistency of Algorithm 2, we primarily use the proof techniques as laid down in the work of Rohe et al. (2011) which were originally used to analyse the spectral clustering algorithm under the high-dimensional block model.

Finally, we analyse our consistency result in some special cases of the block model and validate our findings with accompanying experiments.

## 2. Preliminaries

### 2.1 Stochastic Block Model

The Stochastic Block Model (SBM) is perhaps the most popular model for generating graphs with pre-defined partitions (Holland et al., 1983). We consider an undirected, unweighted graph  $G$  with  $n$  nodes. Under SBM, each node of the graph  $G$  is assigned to one of  $k$  clusters or blocks via the membership matrix  $Z \in \{0, 1\}^{n \times k}$ . Each of the  $n$  rows of  $Z$  has exactly one 1 corresponding to the block to which it belongs i.e.,  $Z_{ig} = 1$  if and only if the node  $i$  belongs to block  $g$ . The probability of the presence of an edge between any two vertices in

the graph is given by the entries of the SBM adjacency matrix  $\mathcal{W}$  defined as

$$\mathcal{W} = ZBZ^T, \quad (1)$$

where  $B \in [0, 1]^{k \times k}$  is the block matrix, whose entry  $B_{gh}$  gives the probability of an edge between nodes of cluster  $g$  and cluster  $h$ .  $B$  is full rank and symmetric. The diagonal entries of  $\mathcal{W}$  are set to zero i.e.,  $\mathcal{W}_{ii} = 0$  for  $1 \leq i \leq n$  to prevent self edges. From  $\mathcal{W}$ , we define the degree matrix  $D$  and the normalized Laplacian matrix  $\mathcal{L}$  as follows.

$$\mathcal{D} = \sum_k \mathcal{W}_{ik}, \quad (2)$$

and

$$\mathcal{L} = \mathcal{D}^{-1/2} \mathcal{W} \mathcal{D}^{-1/2}. \quad (3)$$

To generate a random graph with SBM, we sample a random adjacency matrix  $W$  from it's population version,  $\mathcal{W}$ . Let  $D$  and  $L$  represent the corresponding degree matrix and the normalized Laplacian for the sampled graph. Using Davis-Kahan theorem, it can be shown that the eigenvectors of  $L$  and  $\mathcal{L}$  converge asymptotically as  $n$  becomes large. This is important because the spectral clustering algorithm relies on the eigenvectors of the sampled graph Laplacian  $L$  to estimate the node membership  $Z$ .

Now, we borrow a result (Rohe et al., 2011) that shows the conditions for convergence of the eigenvectors the leading  $k$  eigenvalues (in absolute sense) of  $L$  and  $\mathcal{L}$ . Before this, we define the parameter  $\tau$  as follows.

$$\tau_n = \min_{1 \leq i \leq n} \mathcal{D}_{ii}^{(n)} / n. \quad (4)$$

Here,  $\tau_n$  is the ratio between the minimum expected degree  $\mathcal{D}_{ii}^{(n)}$  of nodes in the graph and the maximum possible degree  $n$ . As such,  $\tau_n$  indicates the level of sparsity in the graph.

**Theorem 2.1.1 (Convergence of Eigenvalues and Eigenvectors)** *Let  $W^{(n)} \in \{0, 1\}^{n \times n}$  be a sequence of random adjacency matrices sampled from the SBM with population matrices  $\mathcal{W}^{(n)}$ . Let  $L^{(n)}$  and  $\mathcal{L}^{(n)}$  be the corresponding graph Laplacian matrices. Let  $X^{(n)}, \mathcal{X}^{(n)} \in \mathbb{R}^{n \times k_n}$  be the matrices that contain the eigenvectors corresponding to the leading  $k_n$  eigenvalues of  $L^{(n)}$  and  $\mathcal{L}^{(n)}$  in absolute sense, respectively. Let  $\bar{\lambda}_{k_n}$  be the least non-zero eigenvalue of  $\mathcal{L}^{(n)}$ . Let the following two assumptions be satisfied.*

**Assumption 1 (Eigengap)**  $n^{-1/2}(\log n)^2 = \mathcal{O}(\bar{\lambda}_{k_n}^2)$

**Assumption 2 (Sparsity)**  $\tau_n^2 > 2/\log n$

*Then, for some sequence of orthonormal matrices  $O^{(n)}$ , the following statements hold almost surely.*

$$\|X^{(n)} - \mathcal{X}^{(n)} O^{(n)}\|_F^2 = o\left(\frac{(\log n)^2}{n \bar{\lambda}_{k_n}^4 \tau_n^4}\right).$$

---

**Algorithm 1** Spectral Clustering

---

**Input:** Graph Laplacian matrix  $L$ , number of clusters  $k$

1. Compute  $X \in \mathbb{R}^{n \times k}$  containing the eigenvectors corresponding to  $k$  leading eigenvalues (in absolute sense) of  $L$ .
2. Treating each row of  $X$  as a point in  $\mathbb{R}^k$ , run  $k$ -means. From the result of  $k$ -means, form the membership matrix  $\hat{Z} \in \{0, 1\}^{n \times k}$  assigning each node to a cluster.

**Output:** Estimated membership matrix  $\hat{Z}$ .

---

**Proof** Theorem 2.1.1 is a special case of Theorem 2.2 from Rohe et al. (2011). The result follows by setting  $S_n = [\lambda_{k_n}^2/2, 1]$  and  $\delta_n = \delta'_n = \bar{\lambda}_{k_n}^2/2$ .  $\blacksquare$

While Assumption 1 ensures that the eigengap of  $\mathcal{L}$  is high enough to enable the separability of the  $k$  clusters, assumption 2 puts a lower bound on the sparsity level of the graph. Under these two assumptions, Theorem 2.1.1 bounds the Frobenius norm of the difference between the top  $k$  eigenvectors of the population and sampled versions of the graph Laplacian.

## 2.2 Spectral Clustering

In the previous subsection, we have seen that the leading eigenvectors of  $L$  and  $\mathcal{L}$  are closely related. Intuitively this makes sense, as the sampled version of the Laplacian  $L$  is only a perturbed version of the population Laplacian  $\mathcal{L}$ . Spectral clustering operates on the  $k$  leading eigenvectors of  $L$  i.e. the matrix  $X$ . Each row of  $X$  is taken as the  $k$ -dimensional spectral embedding of the corresponding node, and  $k$ -means is performed on the new data points to retrieve the cluster membership matrix  $Z$ . The spectral clustering algorithm we consider is listed in Algorithm 1.

Notice that the  $k$ -means is performed on the rows of the matrix  $X$  in Algorithm 1. For this to result in the  $k$  distinct clusters of the SBM, the rows belonging to nodes in different clusters must be ‘well-separated’ while the rows belonging to nodes in the same cluster must be closely spaced. This property of  $X$  becomes evident from Theorem 2.2.1 which follows from the work of Rohe et al. (2011).

**Theorem 2.2.1 (Separability of Clusters)** *Consider a SBM with  $k$  blocks. Let  $\mathcal{L}$  be the population version of the graph Laplacian as defined in (3). Let  $\mathcal{X} \in \mathbb{R}^{n \times k}$  be the matrix containing the eigenvectors corresponding to  $k$  nonzero eigenvalues of  $\mathcal{L}$ . Let  $P$  be the number of nodes in the largest block i.e.  $P = \max_{1 \leq j \leq k} (Z^T Z)_{jj}$ . Then the following statements are true.*

1. *There exists a matrix  $\mu \in \mathbb{R}^{k \times k}$  such that  $Z\mu = \mathcal{X}$ .*
2.  *$\mathcal{X}_{i*} = \mathcal{X}_{j*} \Leftrightarrow Z_{i*} = Z_{j*}$  i.e.  $\mu$  is invertible.*
3.  *$\|\mathcal{X}_{i*} - \mathcal{X}_{j*}\|_2 \geq \sqrt{2/P}$  for any  $Z_{i*} \neq Z_{j*}$ .*

**Proof** Statements 1 and 2 of Theorem 2.2.1 follow from Lemma 3.1 from Rohe et al. (2011). Statement 3 is equivalent to Statement D.3 from the proof of Lemma 3.2 in Rohe

et al. (2011). ■

Here,  $\mathcal{X}_{i*}$  denotes the  $i^{\text{th}}$  row of  $X$ . From Theorem 2.2.1, it is evident that performing  $k$ -means on the rows of  $\mathcal{X}$  would retrieve the block membership of all the nodes in the graph exactly. However, the matrix  $\mathcal{X}$  is hidden, and only its sampled version,  $X$  can be accessed. But by theorem 2.1.1, we have that  $X$  is a close approximation of  $\mathcal{X}$  for large  $n$ . As Algorithm 1 performs  $k$ -means on  $X$ , the estimated membership matrix  $\hat{Z}$  should be close to the true membership matrix  $Z$ .

### 2.3 Graph Filtering

As in Algorithm 1, extracting the top  $k$  eigenvectors of the Laplacian is a key step in the spectral clustering algorithm. In graph signal processing jargon, this can be viewed as extracting the  $k$  lowest frequencies or Fourier modes of the graph Laplacian. This interpretation allows us to use the fast graph filtering approach (Tremblay et al., 2016b; Ramasamy and Madhow, 2015) to speed up the computation. We briefly describe this here.

A graph signal  $y \in \mathbb{R}^n$  is a mapping from vertex set  $V$  of a graph  $G$  to  $\mathbb{R}$ . Let the eigen decomposition of the graph Laplacian be  $L = U\Lambda U^T$ . The graph Fourier transform of signal  $y$  is  $\hat{y} = U^T y$ . The entries of  $\hat{y}$  give the  $n$  Fourier modes of the graph signal  $y$ . Assuming that the rows of  $U$  are ordered in the decreasing order (in absolute value) of the corresponding eigenvalues, the top  $k$  Fourier modes of  $y$  can be obtained by  $\hat{y}_k = X^T y$  where  $X \in \mathbb{R}^{n \times k}$  is the matrix whose columns are the top  $k$  eigenvectors of  $U$ .

Analogous to the classical filtering theory in signal processing, graph filtering can be thought of as an operation to ‘filter’ out specific frequency components of the graph signal. A graph filter function  $h$  is defined over  $[-1, 1]$ , the range of eigenvalues of the normalized graph Laplacian. The filter operator in the graph domain,  $h(\Lambda)$  is a diagonal matrix defined as  $h(\Lambda) := \text{diag}(h(\lambda_1), \dots, h(\lambda_n))$  where  $\lambda_1, \dots, \lambda_n$  are the eigenvalues of  $L$  ordered in the decreasing order of absolute value. The equivalent filter operator in the spectral domain,  $H \in \mathbb{R}^{n \times n}$  is defined as  $H := U h(\Lambda) U^T$ .

To extract the top  $k$  Fourier modes of a graph signal, we use an ideal low-pass filter defined as

$$h_{\lambda_k}(\lambda) = \begin{cases} 1 & \text{if } |\lambda| \geq |\lambda_k| \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The result of graph signal  $y$  filtered through  $h_{\lambda_k}$  is given by  $y_{\lambda_k} = U h_{\lambda_k}(\Lambda) U^T y = X X^T y$ . Obviously, filtering a graph signal with the ideal filter as given in (5) is just as complex as computing the eigen decomposition of the graph Laplacian. Now we shall see how one can use a polynomial approximation to the ideal filter to speed up the computation.

We define  $\tilde{h}_{\lambda_k}(\lambda) := \sum_{\ell=0}^p \alpha_{\ell} \lambda^{\ell}$ , an order  $p$  polynomial, to be the non-ideal approximation of the filter  $h_{\lambda_k}(\lambda)$ . The filter operator in spectral domain,  $\tilde{H}_{\lambda_k}$  can be computed as

$$\tilde{H}_{\lambda_k} = U \tilde{h}_{\lambda_k}(\Lambda) U^T = \sum_{\ell=0}^p \alpha_{\ell} L^{\ell}. \quad (6)$$

The signal  $y$  filtered by  $\tilde{H}_{\lambda_k}$  can be computed as  $\tilde{y}_k = \sum_{\ell=0}^p \alpha_{\ell} L^{\ell} y$ , which does not required the eigen decomposition of  $L$ . Moreover, it only involves computing  $p$  matrix-vector multiplications.

---

**Algorithm 2** Spectral Clustering via Graph Filtering

---

**Input:** Graph Laplacian matrix  $L$ , number of clusters  $k$ , number of dimensions  $d$ , polynomial order  $p$ .

1. Estimate  $\lambda_k$  of  $L$ .
2. Compute the order- $p$  polynomial approximation,  $\tilde{h}_{\lambda_k}$  of the ideal filter  $h_{\lambda_k}$ .
3. Construct an i.i.d Gaussian random matrix  $R \in \mathbb{R}^{n \times d}$  whose entries have a mean 0 and variance  $1/d$ .
4. Compute the approximate embedding matrix  $\tilde{X}_R = \tilde{H}_{\lambda_k} R = \sum_{l=0}^p \alpha_l L^l R$ .
5. Treating each row of  $\tilde{X}_R$  as a point in  $\mathbb{R}^d$ , run  $k$ -means. From the result of  $k$ -means, form the membership matrix  $\hat{Z} \in \{0, 1\}^{n \times k}$  assigning each node to a cluster.

**Output:** Estimated membership matrix  $\hat{Z}$ .

---

## 2.4 The Algorithm

The algorithm that we use for our subsequent analysis is outlined in 1.

## 3. Analysis of Spectral Clustering via Graph Filtering

In this section, we lay down the building blocks that make up the algorithm as outlined in Section 2.4. In 3.1 we shall see how a compressed spectral embedding can be computed with graph filtering and prove that the compressed embedding is still a close approximation of the SBM's population version of graph Laplacian. In Section 3.2 we show the effect of using the fast graph filtering technique to compute the compressed embedding. In Section 3.3 we deal with the estimation of the  $k^{th}$  eigenvalue of the graph Laplacian without resorting to eigen decomposition.

### 3.1 Compressed Spectral Embedding

From Algorithm 1, it seems that we need the matrix  $X$  containing the  $k$  most significant eigenvectors of  $L$ , in order to retrieve the clusters. Since we only use the rows of  $X$  as data points for the subsequent  $k$ -means step, we only need a distance preserving embedding of the rows of  $X$ . In this section, we shall see how such an embedding can be obtained through the result of filtering random graph signals. The technique used is similar to that of Tremblay et al. (2016b), except that we employ slightly stricter assumptions to help in proving consistency results.

Consider the matrix  $R \in \mathbb{R}^{n \times d}$  whose entries are independent Gaussian random variables with mean 0 and variance  $1/d$ . Define  $X_R := H_{\lambda_k} R = U h_{\lambda_k}(\Lambda) U^T R = X X^T R$  whose  $d$  columns contain the result of filtering the corresponding  $d$  columns of  $R$  using the filter  $h_{\lambda_k}$ . In the following theorem, we show that the rows of  $X_R$  form an  $\epsilon$ -approximate distance preserving embedding of the rows of  $X$  for sufficiently large  $d$ . Hence,  $X_R$  is an approximate spectral embedding for the graph Laplacian  $L$ , that is 'compressed' to  $d$  dimensions. To analyse the effect of the compressed embedding on the true cluster centres, i.e. the  $k$  unique rows of  $\mathcal{X}$ , we define the matrix  $\mathcal{X}_R := \mathcal{X} O X^T R$  where  $O$  is the orthonormal rotation matrix as in Theorem 2.1.1. We aim to show that the separability of the true cluster centres is still ensured under the compressed embedding.

**Theorem 3.1.1 (Convergence and Separability under Compressed Spectral Embedding)**

For the sequence of adjacency matrices as defined in Theorem 2.1.1, define  $P_n = \max_{1 \leq j \leq k_n} (Z^T Z)_{jj}$  to be the sequence of populations of the largest block. Let  $X_R^{(n)}, \mathcal{X}_R^{(n)} \in \mathbb{R}^{n \times d_n}$  be the compressed embeddings as defined before for the rows of matrices  $X^{(n)}, \mathcal{X}^{(n)} \in \mathbb{R}^{n \times k_n}$  as defined in Theorem 2.1.1. Let  $\epsilon_1 \in ]0, 1]$  and  $\beta > 0$  be given. If

$$d_n > \frac{4 + 2\beta}{\epsilon_1^2/2 - \epsilon_1^3/3} \log(n + k_n),$$

then with probability at least  $1 - n^{-\beta}$ , we have the following under the assumptions of Theorem 2.1.1.

$$\|X_R^{(n)} - \mathcal{X}_R^{(n)}\|_F^2 = o\left(\frac{(\log n)^2}{n \bar{\lambda}_{k_n}^4 \tau_n^4}\right).$$

Also,

$$\|\mathcal{X}_{R_{i*}}^{(n)} - \mathcal{X}_{R_{j*}}^{(n)}\|_2 \geq (1 - \epsilon_1) \sqrt{2/P_n}$$

for any  $Z_{i*} \neq Z_{j*}$ , where  $i, j \in \{1, \dots, n\}$ .

**Proof** See Appendix A ■

Theorem 3.1.1 is analogous to the theorems on convergence (Theorem 2.1.1) and separability (Theorem 2.2.1) of the spectral clustering algorithm. It ensures that the approximate spectral embedding  $X_R^{(n)}$  converges to the corresponding population version  $\mathcal{X}_R^{(n)}$  while still ensuring that the true clusters remain separable.

**3.2 Efficient Computation via Fast Graph Filtering**

In this section, we define an additional level of approximation for the spectral embedding using the fast graph filtering technique as discussed in Section 2.3.

We define  $\tilde{X}_R := \tilde{H}_{\lambda_k} R = \sum_{\ell=0}^p \alpha_\ell L^\ell R$  to be the output of approximate filtering of the columns of  $R$  where  $R \in \mathbb{R}^{n \times d}$  whose entries are independent Gaussian random variables with mean 0 and variance  $1/d$ . Aim is to bound this result of approximate filtering  $\tilde{X}_R$  with respect to the result of ideal filtering i.e.  $X_R$ . The following theorem establishes this in terms of the accuracy of the polynomial approximation.

**Lemma 3.2.1 (Bounding the Approximate Filtering Error)** For the sequence of adjacency matrices as defined in Theorem 2.1.1, let  $\tilde{X}_R^{(n)} \in \mathbb{R}^{n \times d_n}$  be the approximation for  $X_R^{(n)}$  obtained using the polynomial filter  $\tilde{h}_{\lambda_{k_n}}$  (instead of the ideal filter  $h_{\lambda_{k_n}}$ ). Let  $\sigma(L^{(n)})$  be the spectrum of the sampled graph Laplacian  $L^{(n)}$ . Define the maximum absolute error in the polynomial approximation as

$$e_n = \max_{\lambda \in \sigma(L^{(n)})} |\tilde{h}_{\lambda_{k_n}}(\lambda) - h_{\lambda_{k_n}}(\lambda)|.$$

Let  $\epsilon_2 \in ]0, 1]$  be given. Then with probability at least  $1 - e^{-nd_n(\epsilon_2^2 - \epsilon_2^3)/4}$  we have

$$\|\tilde{X}_R^{(n)} - X_R^{(n)}\|_F^2 \leq (1 + \epsilon_2) n^2 e_n^2.$$

**Proof** See Appendix A ■

Lemma 3.2.1 shows that in order to achieve a fixed error bound between  $X_R$  and  $\tilde{X}_R$ , the polynomial approximation must be ‘increasingly accurate’ as  $n$  grows large. This means that the order  $p_n$  of the polynomial  $\tilde{h}_{\lambda_{k_n}}$  must increase with  $n$ .

### 3.3 Estimation of $\lambda_k$

In the previous section, we have seen that the approximate compressed embedding obtained from fast graph filtering is close to the ideal filtered embedding only when the error in the polynomial approximation is small, particularly at the eigenvalues of the sampled graph Laplacian  $L$ . Designing such a polynomial would necessitate knowing the value of  $\lambda_k$ . In this section, we explain how that can be done without having to do the eigen decomposition of  $L$ . First, we state the following Lemma which bounds the output of fast graph filtering,  $\tilde{X}_R$ .

**Lemma 3.3.1 (Estimation of  $\lambda_k$ )** *For the  $\tilde{X}_R^{(n)}$ ,  $e_n$  and  $\epsilon_2$  given in Lemma 3.2.1, with probability at least  $1 - e^{-nd_n(\epsilon_2^2 - \epsilon_2^3)/4}$  we have*

$$(1 - \epsilon_2)k_n - 2(1 + \epsilon_2)k_n e_n \leq \frac{1}{n} \|\tilde{X}_R^{(n)}\|_F^2 \leq (1 + \epsilon_2)(k_n + 2k_n e_n + n e_n^2).$$

**Proof** See Appendix A. ■

For  $(2k e_n + n e_n^2) = o(1)$ , Lemma 3.3.1 shows that the output of fast graph filtering,  $\tilde{X}_R$  is tightly concentrated around  $k$ , upon normalization by  $n$ . This can be used to estimate  $|\lambda_k|$  by a dichotomic search in the range  $[0, 1]$  as explained in Puy et al. (2016). The basic idea is to make a coarse initial guess on  $|\lambda_k|$  in the interval  $[0, 1]$ , compute  $\tilde{X}_R$  with the current estimate, and iteratively refine the estimate by comparing  $\frac{1}{n} \|\tilde{X}_R\|_F^2$  with  $k$ .

Before we move on to proving the consistency of Algorithm 2, let us summarise the results from the previous sections. We have a tractable way to estimate  $|\lambda_k|$  without the eigen decomposition of  $L$ . Through Lemma 3.2.1, we know that the resultant approximate embedding will be close to the ideal compressed embedding, for reasonably accurate polynomial approximation of the ideal filter. Through Theorem 3.1.1, we showed that a compressed embedding of the  $k$  leading eigenvectors of  $L$  converge to the corresponding embedding on  $\mathcal{L}$ . We also showed that the data points corresponding to different clusters are still separable under such an embedding.

## 4. Consistency of Algorithm SC-GF

### 4.1 Deriving the Error Bound

Once we get the approximate spectral embedding of the  $n$  nodes of the graph in the form of  $\tilde{X}_R$ , we perform  $k$ -means with the rows of  $\tilde{X}_R$  as data points in  $\mathbb{R}^d$ . Let  $c_1, \dots, c_n \in \mathbb{R}^d$  be the centroids corresponding to the  $n$  rows of  $\tilde{X}_R$ , out of which only  $k$  are unique. The  $k$  unique centroids correspond to the centres of the  $k$  clusters. Note that the true cluster centres correspond to the rows of  $\mathcal{X}_R$ , and Theorem 3.1.1 ensures that they are separable from each other. Hence, we say that a node  $i$  is correctly clustered if its  $k$ -means cluster center  $c_i$  is closer to its true cluster center  $\mathcal{X}_{R_{i^*}}$  than it is to any other center  $\mathcal{X}_{R_{j^*}}$ , for



$j \neq i$ . In the following Lemma, we lay down the sufficient condition for correctly clustering a node  $i$ .

**Lemma 4.1.1 (Sufficient Condition for Correct Clustering)** *Let  $c_1^{(n)}, \dots, c_n^{(n)} \in \mathbb{R}^{d_n}$  be the centroids resulting from performing  $k_n$ -means on the rows of  $\tilde{X}_R^{(n)}$ . For  $P_n$  and  $\epsilon_1$  as defined in Theorem 3.1.1,*

$$\|c_i^{(n)} - \mathcal{X}_{R_{i*}}^{(n)}\|_2 < (1 - \epsilon_1) \frac{1}{\sqrt{2P_n}} \Rightarrow \|c_i^{(n)} - \mathcal{X}_{R_{i*}}^{(n)}\|_2 < \|c_i^{(n)} - \mathcal{X}_{R_{j*}}^{(n)}\|_2$$

for any  $z_i \neq z_j$ .

**Proof** See Appendix B. ■

Following the analysis in (Rohe et al., 2011), we define the set of misclustered vertices  $\mathcal{M}$  as containing the vertices that do not satisfy the sufficient condition in Lemma 4.1.1.

$$\mathcal{M} = \left\{ i : \|c_i^{(n)} - \mathcal{X}_{R_{i*}}^{(n)}\|_2 \geq (1 - \epsilon_1) \frac{1}{\sqrt{2P_n}} \right\}$$

Now that we have the definition for misclustered vertices, we analyse the performance of  $k$ -means. Let the matrix  $C \in \mathbb{R}^{n \times k}$  be the result of  $k$ -means clustering where the  $i$ th row,  $c_i$  is the centroid corresponding to the  $i$ th vertex.  $C \in \mathcal{C}_{n,k}$  where  $\mathcal{C}_{n,k}$  represents the family of matrices with  $n$  rows out of which only  $k$  are unique.  $C$  can be defined as

$$C = \arg \min_{M \in \mathcal{C}_{n,k}} \|M - \tilde{X}_R\|_F^2.$$

The next theorem bounds the number of misclustered vertices, that is the size of the set  $\mathcal{M}$ .

**Theorem 4.1.2 (Bound on the number of Misclustered Vertices)**

$$|\mathcal{M}| = o\left(P_n \left( \frac{(\log n)^2}{n \bar{\lambda}_{k_n}^4 \tau_n^4} + n^2 e_n^2 \right)\right) \quad (7)$$

**Proof** See Appendix B. ■

## 4.2 Consistency in Special Cases

In this section, we try to clarify the bound given in Theorem 4.1.2 for special cases of the SBM.

We consider a simplified SBM with four parameters  $k, q, r$  and  $s$  with  $k$  blocks each of which contains  $s$  nodes so that the total number of vertices in the graph,  $n = ks$ . The probability of an edge between two vertices of the same block is given by  $q + r \in [0, 1]$  and that of different blocks is given by  $r \in [0, 1]$ . For the simplified SBM, the population of the largest block,  $P_n = s$ . The smallest non-zero eigenvalue of the sampled graph Laplacian  $L$  is given by

$$\bar{\lambda}_{k_n} = \frac{1}{k(r/q) + 1},$$

and the parameter  $\tau_n$  as defined in (4) is given by (Rohe et al., 2011)

$$\tau_n = q/k + r.$$

Substituting these into the expression in (7), we the proportion of the misclustered vertices for the simplified SBM.

$$\frac{|\mathcal{M}|}{n} = o\left(\frac{k^3}{n}(\log n)^2 + \frac{n^2}{k}e_n^2\right). \quad (8)$$

For weak consistency, we need  $\lim_{n \rightarrow \infty} \frac{|\mathcal{M}|}{n} = 0$ . From (8), the condition on the number of clusters for weak consistency is  $k = o(n^{1/3}/(\log n)^{2/3})$  and the worst case condition on the polynomial approximation error is  $e_n = o(n^{-5/6}(\log n)^{1/3})$ .

## 5. Experiments

We perform experiments on the simplified four parameter SBM presented in Section 4.2. For polynomial approximation of the ideal filter, we use Chebyshev polynomials with Jackson damping coefficients (Di Napoli et al., 2016).

In our first experiment, we analyse the error rate for the SC-GF algorithm for fixed number of clusters as the number of nodes is increased. As expected, the proportion of misclustered vertices,  $\frac{|\mathcal{M}|}{n}$  tends to zero as  $n$  grows large. However, for the case of high polynomial error ( $e_n = \text{and } p =$ ) we see that the error rate diverges. This validates the presence of  $e_n$  in (8).

In our second experiment, we analyse the effect of the polynomial error  $e_n$  in finer detail, by fixing all the other variables,  $n$ ,  $k$ ,  $q$  and  $r$ . From (8) the proportion of misclustered vertices should grow linearly with the squared polynomial error  $e_n^2$ . From Figure 2, this behaviour is evident.

## 6. Conclusion

By Theorem 3.1.1, we prove the fundamental conditions required for the consistency of the spectral clustering algorithm via graph filtering, namely separability and convergence. By Theorem 4.1.2, we have shown that the algorithm can retrieve the planted clusters in a stochastic block model consistently, and derive a bound on the number of misclustered vertices. Through Lemma 3.2.1 and Lemma 3.3.1, we quantify the maximum tolerable filtering error for the algorithm to succeed.

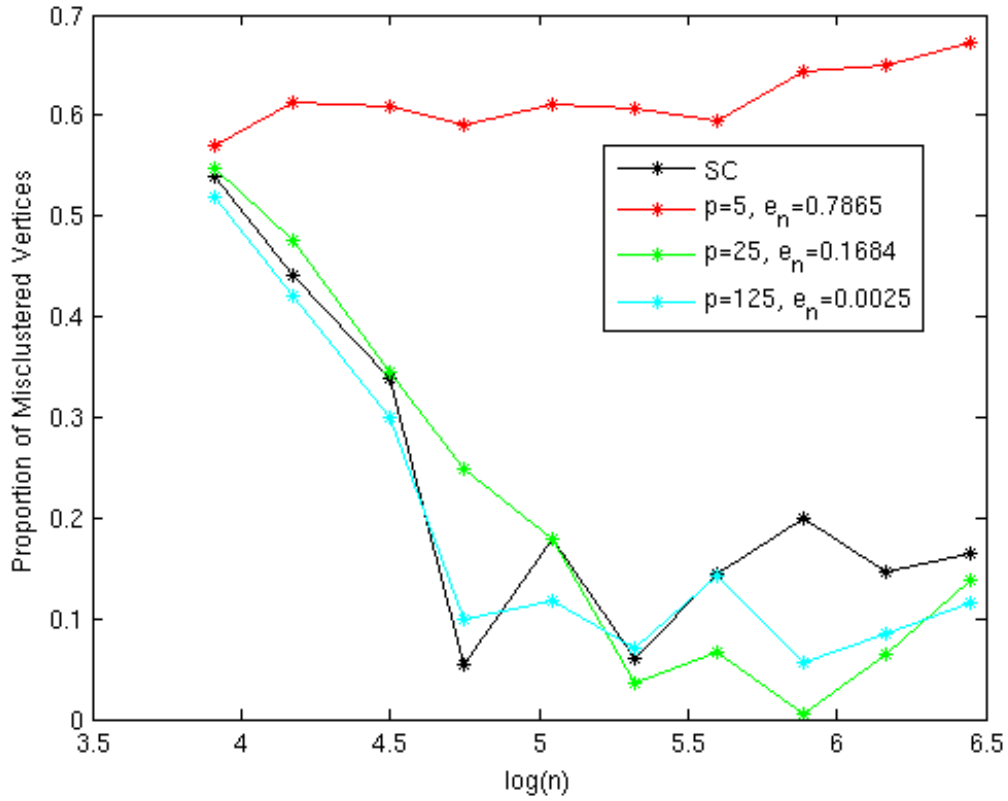


Figure 1: Proportion of misclustered vertices plotted against the logarithm of total number of vertices.  $q = 0.3$  and  $r = 0.1$ . The polynomial order  $p$  is set to 5, 25 and 125 for the three curves pertaining to SC-GF algorithm. The corresponding polynomial error  $e_n$  is shown in the legend.

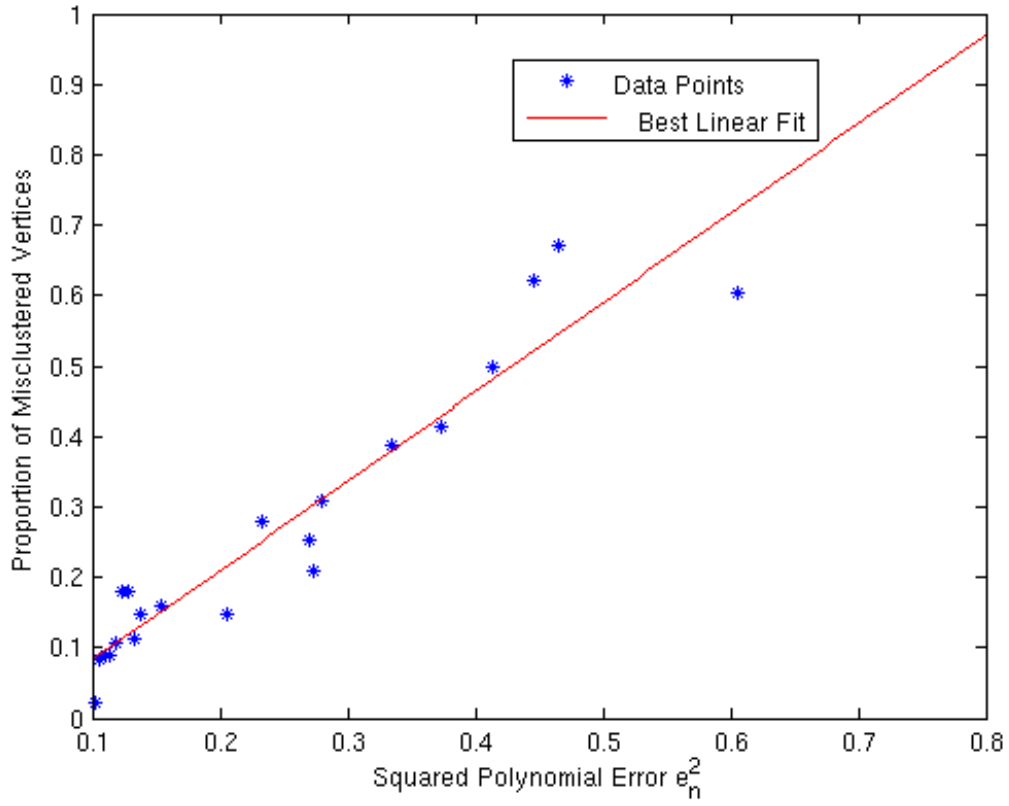


Figure 2: Proportion of misclustered vertices plotted against the squared polynomial error,  $e_n^2$ .  $q = 0.3$  and  $r = 0.1$ . The polynomial order  $p$  is varied from 5 to 25 linearly.

## References

- Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.
- Anna Choromanska, Tony Jebara, Hyungtae Kim, Mahesh Mohan, and Claire Monteleoni. Fast spectral clustering via the nystrom method. In *International Conference on Algorithmic Learning Theory*, pages 367–381. Springer, 2013.
- Edoardo Di Napoli, Eric Polizzi, and Yousef Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 23(4):674–692, 2016.
- Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.
- Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.
- Alex Gittens, Prabhanjan Kambadur, and Christos Boutsidis. Approximate spectral clustering via randomized sketching. *Ebay/IBM Research Technical Report*, 2013.
- Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.
- Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
- Jing Lei, Alessandro Rinaldo, et al. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.
- Mu Li, Xiao-Chen Lian, James T. Kwok, and Bao-Liang Lu. Time and space efficient spectral clustering via column sampling. In *CVPR*, 2011.
- Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.
- Gilles Puy, Nicolas Tremblay, Rémi Gribonval, and Pierre Vandergheynst. Random sampling of bandlimited signals on graphs. *Applied and Computational Harmonic Analysis*, 2016.
- Dinesh Ramasamy and Upamanyu Madhow. Compressive spectral embedding: sidestepping the svd. In *Advances in Neural Information Processing Systems*, pages 550–558, 2015.
- Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, pages 1878–1915, 2011.
- Tomoya Sakai and Atsushi Imiya. Fast spectral clustering with random projection and sampling. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 372–384. Springer, 2009.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Nicolas Tremblay, Gilles Puy, Pierre Borgnat, Pierre Vandergheynst, et al. Accelerated spectral clustering using graph filtering of random signals. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4094–4098. IEEE, 2016a.

Nicolas Tremblay, Gilles Puy, Rémi Gribonval, and Pierre Vandergheynst. Compressive spectral clustering. In *Machine Learning, Proceedings of the Thirty-third International Conference (ICML 2016), June*, pages 20–22, 2016b.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

## Appendix A. Proofs for Theorems in Section 3

### A.1 Proof of Theorem 3.1.1

**Proof** For the sake of compactness, we omit the superscript  $(n)$  for the sequences of matrices, as the analysis is valid at every  $n$ .

By Theorem 2.2.1, there are at most  $k$  unique rows out of the  $n$  rows of the matrix  $\mathcal{X}$ , while the  $n$  rows of the matrix  $X$ , can potentially be unique. The same inference can be made for the matrices  $\mathcal{X}OX^T$  and  $XX^T$ , where  $O$  is the orthonormal matrix from Theorem 2.1.1.

Treating the combined  $n + k$  unique rows of the two matrices as data points in  $\mathbb{R}^k$ , we can use the Johnson-Lindenstrauss Lemma to approximately preserve the pairwise Euclidian distances between any two rows up to a factor of  $\epsilon_1$ . Applying Theorem 1.1 from Achlioptas (2003), if  $d_n$  is larger than

$$\frac{4 + 2\beta}{\epsilon_1^2/2 - \epsilon_1^3/3} \log(n + k),$$

then with probability at least  $1 - n^{-\beta}$ , we have

$$(1 - \epsilon_1) \|\mathcal{X}_{i*}OX^T - \mathcal{X}_{j*}OX^T\|_2 \leq \|\mathcal{X}_{Ri*} - \mathcal{X}_{Rj*}\|_2 \leq (1 + \epsilon_1) \|\mathcal{X}_{i*}OX^T - \mathcal{X}_{j*}OX^T\|_2 \quad (9)$$

for any  $Z_{i*} \neq Z_{j*}$ ,

$$(1 - \epsilon_1) \|X_{i*}X^T - X_{j*}X^T\|_2 \leq \|X_{Ri*} - X_{Rj*}\|_2 \leq (1 + \epsilon_1) \|X_{i*}X^T - X_{j*}X^T\|_2$$

and

$$(1 - \epsilon_1) \|X_{i*}OX^T - \mathcal{X}_{j*}X^T\|_2 \leq \|X_{Ri*} - \mathcal{X}_{Rj*}\|_2 \leq (1 + \epsilon_1) \|X_{i*}OX^T - \mathcal{X}_{j*}X^T\|_2 \quad (10)$$

where  $i, j \in \{1, \dots, n\}$ .

Combining the inequality on the left side of (9) with Statement 3 of Theorem 2.2.1, we get

$$\begin{aligned}\|\mathcal{X}_{Ri*} - \mathcal{X}_{Rj*}\|_2 &\geq (1 - \epsilon_1)\|\mathcal{X}_{i*}OX^T - \mathcal{X}_{j*}OX^T\|_2 \\ &= (1 - \epsilon_1)\|(\mathcal{X}_{i*} - \mathcal{X}_{j*})OX^T\|_2 \\ &= (1 - \epsilon_1)\|\mathcal{X}_{i*} - \mathcal{X}_{j*}\|_2 \\ &\geq (1 - \epsilon)\sqrt{2/P_n}\end{aligned}$$

for any  $Z_{i*} \neq Z_{j*}$ . Since  $X^TX$  is an identity matrix, the rows of  $OX^T$  are orthogonal. Hence, multiplication of a vector by  $OX^T$  from the right does not change the norm. By a similar procedure, combining the inequality on the right side of (10) with Theorem 2.1.1, we get

$$\begin{aligned}\|X_R - \mathcal{X}_R\|_F^2 &= \sum_{i=1}^n \|X_{Ri*} - \mathcal{X}_{Ri*}\|_2^2 \leq (1 + \epsilon_1)^2 \sum_{i=1}^n \|X_{i*}OX^T - \mathcal{X}_{j*}X^T\|_2^2 \\ &= (1 + \epsilon_1)^2 \sum_{i=1}^n \|(X_{i*}O - \mathcal{X}_{j*})X^T\|_2^2 \\ &= (1 + \epsilon_1)^2 \sum_{i=1}^n \|X_{i*}O - \mathcal{X}_{j*}\|_2^2 \\ &= (1 + \epsilon_1)^2 \|X_{i*}O - \mathcal{X}_{j*}\|_F^2 \\ &= o\left(\frac{(\log n)^2}{n\bar{\lambda}_{k_n}^4 \tau_n^4}\right)\end{aligned}$$

■

## A.2 Proof of Lemma 3.2.1

**Proof** Firstly, we note that  $\|R^{(n)}\|_F^2$  is a chi-squared random variable with  $nd_n$  degrees of freedom and mean  $n$ . Using the Chernoff bound on  $\|R^{(n)}\|_F^2$ , we have

$$\Pr\left(\left|\frac{1}{n}\|R^{(n)}\|_F^2 - 1\right| > \epsilon_2\right) \leq e^{-nd_n(\epsilon_2^2 - \epsilon_2^3)/4} \quad (11)$$

Now to bound the difference between the ideal and polynomial filters,

$$\begin{aligned}\|U(\tilde{h}_{\lambda_{k_n}}(\Lambda) - h_{\lambda_{k_n}}(\Lambda))U^T\|_F^2 &= \|\tilde{h}_{\lambda_{k_n}}(\Lambda) - h_{\lambda_{k_n}}(\Lambda)\|_F^2 = \sum_{i=1}^n (\tilde{h}_{\lambda_{k_n}}(\lambda_i) - h_{\lambda_{k_n}}(\lambda_i))^2 \\ &\leq \sum_{i=1}^n e_n^2 = ne_n^2.\end{aligned} \quad (12)$$

Using the result from (11) and (12), we can bound the difference between the ideal and approximate spectral embedding as follows.

$$\begin{aligned}\|\tilde{X}_R^{(n)} - X_R^{(n)}\|_F^2 &= \|\tilde{H}_{\lambda_{k_n}} R - H_{\lambda_{k_n}} R^{(n)}\|_F^2 = \|U(\tilde{h}_{\lambda_{k_n}}(\Lambda) - h_{\lambda_{k_n}}(\Lambda))U^T R^{(n)}\|_F^2 \\ &\leq \|U(\tilde{h}_{\lambda_{k_n}}(\Lambda) - h_{\lambda_{k_n}}(\Lambda))U^T\|_F^2 \|R^{(n)}\|_F^2 \\ &\leq (1 + \epsilon_2)n^2 e_n^2\end{aligned}$$

where the last step follows with a probability of at least  $1 - e^{-nd_n(\epsilon_2^2 - \epsilon_2^3)/4}$ .  $\blacksquare$

### A.3 Proof of Lemma 3.3.1

**Proof** For the sake of compactness, we omit the superscript  $(n)$  for the sequences of matrices, as the analysis is valid at every  $n$ .

From Lemma 3.2.1, we have a bound on the term  $\|\tilde{X}_R - X_R\|_F^2$ . So, we proceed to prove Lemma 3.3.1 by bounding the term  $\|X_R\|_F^2$ . For this, we make use of the fact that the  $k_n$  columns of  $X^{(n)}$  are orthonormal.

$$\|X_R\|_F^2 = \|XX^{(n)T}R\|_F^2 = k_n\|R\|_F^2 \quad (13)$$

Combining (13) with (11), we have the following with probability exceeding  $1 - e^{-nd_n(\epsilon_2^2 - \epsilon_2^3)/4}$ .

$$(1 - \epsilon_2)k_n \leq \frac{1}{n}\|X_R\|_F^2 \leq (1 + \epsilon_2)k_n.$$

Now, we prove the upper bound on  $\tilde{X}_R$ .

$$\begin{aligned}\|\tilde{X}_R\|_F^2 &= \text{tr}(\tilde{X}_R^T \tilde{X}_R) = \text{tr}(R^T U \tilde{h}_{\lambda_k}(\Lambda) U^T U \tilde{h}_{\lambda_k}(\Lambda) U^T R) \\ &= \text{tr}(R^T U (\tilde{h}_{\lambda_k}(\Lambda))^2 U^T R) \\ &= \text{tr}((\tilde{h}_{\lambda_k}(\Lambda))^2 U^T R R^T U) \\ &\leq \text{tr}((\tilde{h}_{\lambda_k}(\Lambda))^2) \text{tr}(U^T R R^T U)\end{aligned} \quad (14)$$

where the last statement follows from the fact that the matrices  $\tilde{X}_R^T \tilde{X}_R$ ,  $(\tilde{h}_{\lambda_k}(\Lambda))^2$  and  $U^T R R^T U$  are non-negative semi-definite.

$$\text{tr}(U^T R R^T U) = \text{tr}(R R^T) = \|R\|_F^2 \leq (1 + \epsilon_2)n. \quad (15)$$

The last statement follows from (11) with a probability of at least  $1 - e^{-nd_n(\epsilon_2^2 - \epsilon_2^3)/4}$ .

Using the definition of the maximum filter error  $e_n$ , we get

$$\text{tr}((\tilde{h}_{\lambda_k}(\Lambda))^2) \leq k(1 + e_n)^2 + (n - k)e_n^2 = k + 2ke_n + ne_n^2. \quad (16)$$

Combining (15) and (16) with (14), we get

$$\frac{1}{n}\|\tilde{X}_R\|_F^2 \leq (1 + \epsilon_2)(k + 2ke_n + ne_n^2). \quad (17)$$



Now we proceed to proving the lower bound.

$$\|\tilde{X}_R\|_F^2 = \|X_R\|_F^2 + \|\tilde{X}_R - X_R\|_F^2 + 2 \operatorname{tr}(X_R^T(\tilde{X}_R - X_R)). \quad (18)$$

$$\begin{aligned} \operatorname{tr}(X_R^T(\tilde{X}_R - X_R)) &= \operatorname{tr}(R^T U h_{\lambda_k}(\Lambda) U^T U (\tilde{h}_{\lambda_k}(\Lambda) - h_{\lambda_k}(\Lambda)) U^T R) \\ &= \operatorname{tr}(h_{\lambda_k}(\Lambda) (\tilde{h}_{\lambda_k}(\Lambda) - h_{\lambda_k}(\Lambda)) U^T R R^T U) \\ &= \operatorname{tr}(h_{\lambda_k}(\Lambda) (\tilde{h}_{\lambda_k}(\Lambda) - h_{\lambda_k}(\Lambda) + e_n I_n) U^T R R^T U) \\ &\quad - \operatorname{tr}(h_{\lambda_k}(\Lambda) e_n I_n U^T R R^T). \end{aligned} \quad (19)$$

Here,  $I_n \in \mathbb{R}^{n \times n}$  is the Identity matrix. By the definition of  $e_n$ , the diagonal entries of  $(\tilde{h}_{\lambda_k}(\Lambda) - h_{\lambda_k}(\Lambda) + e_n I_n)$  are non-negative. Hence, the first term in (19) is non-negative. For the second term we have,

$$\operatorname{tr}(h_{\lambda_k}(\Lambda) e_n I_n U^T R R^T) \leq e_n \operatorname{tr}(h_{\lambda_k}(\Lambda)) \operatorname{tr}(U^T R R^T) = e_n k_n (1 + \epsilon_2) n. \quad (20)$$

In addition, the term  $\|\tilde{X}_R - X_R\|_F^2$  in (18) is non-negative. Combining (20) with (18), we get

$$\frac{1}{n} \|\tilde{X}_R\|_F^2 \geq (1 - \epsilon_2) k_n - 2(1 + \epsilon_2) e_n k_n. \quad (21)$$

Putting together (17) and (21), we prove Lemma 3.3.1.  $\blacksquare$

## Appendix B. Proofs for Theorems in Section 4

### B.1 Proof of Lemma 4.1.1

**Proof** We follow a similar technique as that of Lemma 3.2 in Rohe et al. (2011). Suppose that  $\|c_i^{(n)} - \mathcal{X}_{R_{i*}}^{(n)}\|_2 < (1 - \epsilon_1) \frac{1}{\sqrt{2P_n}}$  for some  $i$ . For any  $z_j \neq z_i$ , we have

$$\begin{aligned} \|c_i^{(n)} - \mathcal{X}_{R_{j*}}^{(n)}\|_2 &\geq \|\mathcal{X}_{R_{i*}}^{(n)} - \mathcal{X}_{R_{j*}}^{(n)}\|_2 - \|c_i^{(n)} - \mathcal{X}_{R_{i*}}^{(n)}\|_2 \geq (1 - \epsilon_1) \sqrt{\frac{2}{P_n}} - (1 - \epsilon_1) \frac{1}{\sqrt{2P_n}} \\ &= (1 - \epsilon_1) \frac{1}{\sqrt{2P_n}} \end{aligned}$$

Here we have used the result of Theorem 3.1.1 on the separability of the rows of  $\mathcal{X}_R^{(n)}$ .  $\blacksquare$

### B.2 Proof of Theorem 4.1.2

**Proof** From the output of the  $k$ -means we have

$$C = \arg \min_{M \in \mathcal{C}_{n,k}} \|M - \tilde{X}_R\|_F^2$$

From Theorem 2.2.1 we know that only  $k$  rows of the matrix  $\mathcal{X}$  are unique out of its  $n$  rows. The same inference can be made about  $\mathcal{X}_R$ . Hence,  $\mathcal{X}_R \in \mathcal{C}_{n,k}$ . By the optimality of  $k$ -means we have

$$\|C - \tilde{X}_R\|_F^2 \leq \|\mathcal{X}_R - \tilde{X}_R\|_F^2 \leq 2\|\mathcal{X}_R - X_R\|_F^2 + 2\|X_R - \tilde{X}_R\|_F^2.$$

Hence

$$\begin{aligned} \|C - \mathcal{X}_R\|_F^2 &\leq \|C - \tilde{X}_R\|_F^2 + \|\tilde{X}_R - \mathcal{X}_R\|_F^2 \\ &\leq \left(2\|\mathcal{X}_R - X_R\|_F^2 + 2\|X_R - \tilde{X}_R\|_F^2\right) + \left(2\|\tilde{X}_R - X_R\|_F^2 + 2\|X_R - \mathcal{X}_R\|_F^2\right) \\ &= 4\|\mathcal{X}_R - X_R\|_F^2 + 4\|X_R - \tilde{X}_R\|_F^2 \end{aligned}$$

From the definition of the misclustered vertices,

$$\begin{aligned} |\mathcal{M}| &\leq \sum_{i \in \mathcal{M}} 1 \leq \frac{2P_n}{(1 - \epsilon_1)^2} \sum_{i \in \mathcal{M}} \|c_i - \mathcal{X}_{R_{i*}}\|_F^2 \\ &\leq \frac{2P_n}{(1 - \epsilon_1)^2} \|C - \mathcal{X}_R\|_F^2 \\ &\leq \frac{2P_n}{(1 - \epsilon_1)^2} \left(4\|\mathcal{X}_R - X_R\|_F^2 + 4\|X_R - \tilde{X}_R\|_F^2\right) \\ &= o\left(P_n \left(\frac{(\log n)^2}{n\bar{\lambda}_{k_n}^4 \tau_n^4} + n^2 e_n^2\right)\right). \end{aligned}$$

The last statement follows from Theorem 3.1.1 and Lemma 3.2.1. ■